**Participants:**

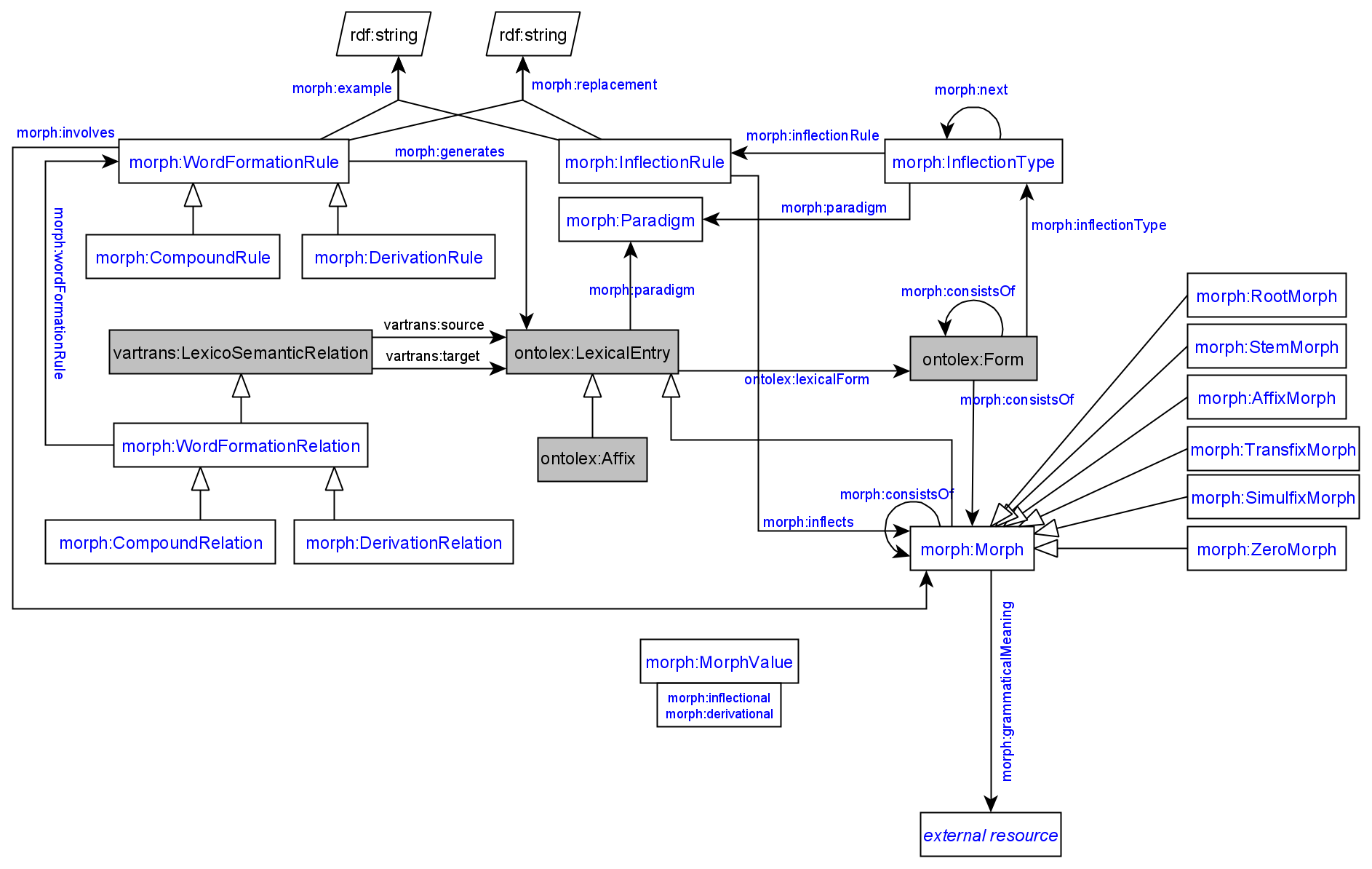
Bettina Klimek (BK)

Christian Chiarcos (CC)

Matteo Pellegrini

Katerina Gkirtzou (KG)

1. **Module draft 4.6**

****

Included adaptations of module draft 4.6

* morph:Morph as subclass of ontolex:LexicalEntry → results in morph:sense and morph:evokes being unnecessary (removed)
  + JMC: senses for morphs and lexical entries might not be the same thing (which they would be if morph:Morph is a subclass of ontolex:LexicalEntry)
* renamed morph:contains to morph:involves
* added object property morph:contains between morph:WordFormationRule and morph:Morph (means that vartrans:source and target can stay restricted to ex. 1 ontolex:LexicalEntry)
* removed object property morph:hasMorphStatus
* morph string values can be represented as lexicalFormwith ontolex:writtenRep
* allomorphs can be represented as otherForm
* morpheme “value” => canonicalForm
* ?rename morph:consistsOf?

BK: Suggestion for providing the morphological status (i.e. inflectional or derivational) for morph:Morph resources with the owl:hasValue object property restriction:

owl:Restriction

owl:onProperty morph:morphValue

owl:hasValue rdf:resource= morph:inflectional OR morph:derivational

morph:morphValue rdfs:subPropertyOf ontolex:lexicalForm

morph:inflection a morph:MorphValue .

morph:derivational a morph:MorphValue .

Example:

<morph#dNV09%3E> rdf:type morph:Morph ;

rdf:type morph:SuffixMorph ;

ontolex:canonicalForm \_:b6571 ;

rdfs:subClassOf morph:MorphValue .

\_:x rdf:type owl:Restriction ;

owl:onProperty morph:morphValue ;

owl:hasValue morph:inflectional .

CC: if the property has only 2 values we can use a boolean

PL: we might not know all languages and some might have more than the 2 values

Adaptations of module draft 4.7 to be included for next telco:

* "lexinfo:termElement lexinfo:inflectionElement", so MorphValue is removed
* remove morph:DerivationRelation (CC in favor of eliminating the redundancy between DerivationRelation subclasses and DerivationRule (would make diagram more readable) because the difference between morph:DerivationRelation and morph:DerivationRule can be expressed in morph:DerivationRule alone
* other proposed changes:
  + merge ontolex:LexicalEntry and ontolex:Affix subclasses, turning ontolex:Affix and morph:Non-Affix subclasses of morph:Morph and sort the morph affix subclasses under ontolex:Affix
  + composition with more than 2 elements resolved by CC by using morph:CompositionalRelation and the decomp vocabulary
    - suggestion: use CompositionalRelation \*only\* to mark the morphological head, modifiers go unmarked
  + example (after call):
    - Blutspendezentrale Blut spenden Zentrale “blood donation center” (GermaNet: <https://www.sfs.uni-tuebingen.de/GermaNet/documents/compounds/split_compounds_from_GermaNet16.0.txt>)

<entry#Blutspendezentrale>

decomp:subterm <entry#Zentrale> ;

decomp:subterm <entry#spenden> ;

decomp:subterm <entry#Blut> ;

rdf:type ontolex:LexicalEntry ;

ontolex:canonicalForm <form#Blutspendezentrale> .

# currently, we only create a rule and rel if there is an interfix involved, so I have to make the following up for this example

<rule#0> morph:generates <entry#Blutspendezentrale> ;

morph:example "Blut + spenden > Zentrale" .

[] rdf:type morph:CompoundRelation;

vartrans:source <entry#Zentrale>;

vartrans:target <entry#Blutspendezentrale>;

morph:wordFormationRule <rule#0>.

1. **Derinet data conversion examples**

[**https://github.com/acoli-repo/acoli-morph/blob/main/uder/Readme.md**](https://github.com/acoli-repo/acoli-morph/blob/main/uder/Readme.md)

# ?a: "Aal\_Nn\_form"

<entry#Aal\_Nn> rdf:type ontolex:LexicalEntry ;

uder:POS "Nn" ;

ontolex:canonicalForm <form#Aal\_Nn\_form> .

<form#Aal\_Nn\_form> ontolex:writtenRep "Aal" .

# ?b: "aalen\_V"

<entry#aalen\_V> rdf:type ontolex:LexicalEntry ;

uder:POS "V" ;

ontolex:canonicalForm <form#aalen\_V\_form> .

<form#aalen\_V\_form> ontolex:writtenRep "aalen" .

# ?c: "NV09>"

<rel#dNV09%3E> rdf:type morph:DerivationRelation ;

vartrans:source <entry#Aal\_Nn> ;

vartrans:target <entry#aalen\_V> .

# derived information

<rel#dNV09%3E>

morph:contains <morph#dNV09%3E> ;

morph:wordFormationRule <rule#dNV09%3E> .

<rule#dNV09%3E> rdf:type morph:DerivationRule ;

morph:generates <entry#aalen\_V> ;

morph:example "Aal\_Nn > aalen\_V" ;

morph:replacement "s/$/en/" .

<morph#dNV09%3E> rdf:type morph:Morph ;

rdf:type morph:SuffixMorph ;

# this is change to the model: if had morph:Affix only

rdf:type ontolex:Affix ;

ontolex:lexicalForm \_:b6571 .

# "-en" is obtained from a string match between ?a and ?b

# we don't know whether these are unique for the rule "NV09>"

\_:b6571 rdf:type ontolex:Form ;

ontolex:writtenRep "-en" .

# this is obtained automatically from comparing ?a (Aal) and ?b (aalen)

<morph#dNV09%3E> morph:PLEASE\_GIVE\_ME\_A\_NAME\_FOR\_CONSTRAINTS "Nn" ;

morph:grammaticalMeaning "V" .

<form#aalen\_V\_form> morph:consistsOf <morph#dNV09%3E> .

**Todo:**

**BK:** look into lexinfo to be reused for MorphValue (CC: we can say "lexinfo:termElement lexinfo:inflectionElement", so MorphValue is not necessary) + create module diagram draft 4.7

**Matteo:** reports about representing compounds in Lila with more than 2 components

**Penny:** shares Greek inflectional data example based on morph module vocabulary

topics for the next telco:

* representing ordering
* creating a property to express <morph#dNV09%3E> morph:PLEASE\_GIVE\_ME\_A\_NAME\_FOR\_CONSTRAINTS "Nn"